Resource-Aware Adaptive Multicasting in a Shared Proxy Overlay Network

Background of the Invention

[0001] This invention relates generally to peer-to-peer overlay networks of proxy servers, and more particularly to proxy overlay networks for streaming live media.

[0002] Wide area networks, such as the Internet, that are used for multicasting streaming media from a plurality of sources to a plurality of end users comprise a network of multiple autonomous systems ("ASs") and an overlay network of proxy servers ("proxies") that function as nodes for multicasting the streaming media. Each AS may host one or more proxy servers which can direct streaming traffic across the borders of the AS to a neighboring AS. Proxy servers may multiplex media streams received from other proxy servers or from media servers, and provide the multiplexed streams to other proxies or to end users. Each AS has a multiplexing capacity which is limited by the processing power of its proxy servers and the local bandwidth limitations of paths between ASs. An AS can also "tunnel" a stream, i.e., pass the stream, from a neighboring AS to another AS. While this uses bandwidth resources at the peering points (communications paths that cross borders between ASs), it does not affect the multiplexing capacity of the AS, which depends primarily on local resources.

[0003] Known multicasting overlay networks create multicast tree structures using peers of proxies that independently enter and leave the network. These tree structures are typically created in known systems so as to minimize the latency between end users and a corresponding media source. This approach, however, does not utilize resource capacity in the most efficient or least costly way. For example, an overlay network typically has to support multiple simultaneous media streams, each with a different source and user population. Known approaches, however, do not appropriately consider multiple media sources and user populations and do not effectively share the resources of the overlay network in an optimum manner.

[0004] Additionally, another problem is that known overlay networks do not minimize operational costs or use resources most effectively. Each AS is an independent entity (such as an Internet service provider, for example), which has to be compensated monetarily for the resources of the AS consumed by the media streams it delivers. Consequently, each AS on which an overlay network operates will charge the overlay network for the bandwidth streamed through it. Additionally, ASs are connected to each other through peering points, each of which has a maximum bandwidth capacity. Any stream that crosses an AS boundary utilizes a portion of the limited capacity of the path as well as of the overlay network. Purchasing, deploying and maintaining a proxy server in an AS is costly, and present overlay networks find it difficult to balance the conflicting goals of minimizing physical resources to minimize costs while avoiding overutilization of network resources and peering points in the face of varying workloads.

[0005] There is a need for method and systems which avoid the foregoing problems of known shared proxy overlay networks for multicasting media streams by affording better utilization of resources and reduced costs while accommodating the needs of users requesting access to streaming media, and it is to these ends that the present invention is directed.

Summary of the Invention

[0006] The invention solves the foregoing and other problems of known proxy overlay networks for streaming media by affording an overlay proxy server network structure that uses proxy-to-proxy negotiations for establishing and optimizing utilization of resources. Proxy servers communicate to establish or restructure the network structure to optimize utilization of network resources through request redirections, load redistributions and load consolidations across autonomous systems. Restructuring may occur dynamically as conditions change.

[0007] A proxy that receives a join request for access may redirect the request to a more suitable proxy that can more efficiently handle the request. Additionally, a proxy which is overloaded may initiate redistribution to send part of its load to a different proxy, and an underutilized proxy may send part of its load to another proxy where it is combined or consolidated with other loads. The invention employs an information exchange messaging protocol the maintains information as to the status of other network resources. This facilitates adaptively shifting loads and responding to requests by changing network structure to afford optimum use of resources. The invention avoids overutilized or underutilized resources, and minimizes end-to-end delays and overall operational costs.

[0008] In one aspect, the invention provides a method of distributing streaming data in a wide area network which comprises an overlay network of proxy servers on autonomous systems in which proxy servers communicate with neighboring proxy servers to identify proxy servers and data paths that optimize utilization of network resources based upon a predetermined relationship that characterizes tensions of the proxy servers and data paths. The proxy servers communicate by exchanging messages; and, in responds to the messages, activate neighboring proxy servers to form a portion of a hierarchial overlay network structure as applies the data stream to a requester.

[0009] In another aspect, the invention provides a method which may activate, in response to communicating between proxy servers, first proxy servers of the overlay network to form a first hierarchial overlay network structure of proxy servers to establish a plurality of data paths through the overlay network to distribute a first data stream from a first data source to a first group of requesters. The method further activates in response to the communicating second proxy servers to form a second hierarchial structure to establish another plurality of data paths through the overlay network to distribute a second data stream from a second data source to a group of requestors. The first and second hierarchial structures may share one or more of the first and second proxy servers.

[0010] In yet another aspect, the invention provides a method of distributing streaming data in a wide area network that optimizes utilization of network resources based upon a predetermined relationship that characterizes the tensions of the proxy servers and data paths by exchanging messages with neighboring proxy servers and utilizing stored information concerning proxy servers status to activate a proxy server to form an optimum data path for supplying a data stream to a requester.

Brief Description of the Drawings

[0011] Figure 1 is a diagrammatic view illustrating a multicasting overlay network in accordance wit the invention, showing the sharing of proxy resources by two media sources;

[0012] Figure 2 is a diagrammatic view illustrating two autonomous systems (ASs), each containing a plurality of proxy servers, and being connected at a peering point;

[0013] Figure 3 comprising Figures 3(a) - (b) illustrates two alternative resource tension graphs which may be utilized by the invention for creating and maintaining multicast tree structures;

[0014] Figure 4 comprising Figures 4(a) - (b) illustrates redirection of a request;

[0015] Figure 5 comprising Figures 5(a) - (b) illustrates load redistribution to avoid overutilization of a resource;

[0016] Figure 6 comprising Figures 6(a) - (b) illustrates load consolidation to avoid underutilization of a resource;

[0017] Figure 7 is a flow chart illustrating a process performed by a proxy in response to a join request;

[0018] Figure 8 comprising Figures 8(a) - (b) illustrate, respectively, request redirection and tunneling processes;

[0019] Figure 9 comprising Figures 9(a) - (c) illustrate an expansion process of a neighborhood through redirection;

[0020] Figure 10 illustrates a process of selecting alternative parents using cost estimation;

[0021] Figure 11 comprising Figures 11(a) - (b) illustrate loop avoidance in join requests; and

[0022] Figure 12 comprising Figures 12(a) - (c) illustrate loop avoidance in redistribution.

Description of Preferred Embodiments

[0023] The invention is particularly adapted for use in a shared proxy overlay network employed for simultaneous multicasting of live media streams and will be described in that context. It will become apparent, however, that this is illustrative of only one utility of the invention.

[0024] Figure 1 illustrates a multicasting overlay network in accordance with the invention in which overlay network proxy resources may be shared to provide access to media streams from different media sources. As shown, the overlay network may comprise a plurality of proxy servers 10 which may be overlaid on a plurality of autonomous systems (ASs) 12-14 to provide access to media streams from media sources S1 and S2. The dotted lines from the media sources and between the proxy servers represent communications paths or links between the overlay network resources for the two media streams. The designations "S1" and "S2" within the circles in the figure, which represent proxy servers, indicate that a proxy server is a shared resource that provides access to both streams by using a portion of its capacity for each stream. The positions of the horizontal lines across the circles, which indicate the proxy servers 10, represent the portion of a capacity of the proxy server utilized by each media stream. As is also shown in the figure, the proxy servers are linked or connected by paths so as to form separate hierarchical multicast tree structures with respect to each media stream. The proxy servers may be located within each AS upon agreement with the operators of the AS. Each AS on which the overlay network operates will typically charge for the bandwidth resources used for the media streamed through it, and the overlay network will consequently incur a corresponding monetary cost. Thus, for example, the streams 20 and 21 from AS 12 to AS 13 will be through a peering point (an inter-AS data path for streaming media as will be described) having a particular bandwidth capacity, and charges will be incurred based upon the amount of the capacity of the peering point used. Typically, the paths between proxy servers within an AS, such as paths 24 and 25 within AS 13, do not incur charges based upon the resources used (once a basic subscription fee is paid by a user) and, therefore, are substantially free. Therefore, it is advantageous to minimize the traffic that flows across AS boundaries to minimize costs and the invention takes this into consideration, in a manner that will be described, in constructing and maintaining a multicast tree structure.

[0025] Figure 2 illustrates the structure of two autonomous systems and their peering point connection in more detail. As shown, autonomous systems AS1 and AS2 may contain a plurality of interconnected proxies. As shown, AS1 may have proxies 30-32 and AS2 may have proxies 33-34. The two ASs are connected at their boundary by a peering point 36. The peering point comprises a communications path or link between the two ASs which has a predetermined maximum bandwidth capacity that may be established by service agreements between the operators of each AS. This may be less than the actual maximum bandwidth capacity of the hardware resources utilized for the peering point. Consequently, any media stream that crosses the peering point boundary 36 between the ASs utilizes the limited resource and incurs a corresponding charge. As noted earlier, the paths between proxies within an AS, such as indicated by the lines between proxies 30-32 in AS1, may be free. Therefore, it is desirable to minimize the amount of traffic that flows across AS boundaries in constructing or maintaining the multicast tree, and the invention takes this cost into consideration in a manner which will be described.

[0026] Each AS in a physical wideband network may contain none or one or more proxy servers. For the purposes of explanation in this specification, all proxy servers within a given AS will be treated as a single proxy server. The role of a proxy server is to multiplex streams it receives parent sources, such as from media sources or other proxies, and to serve the multiplexed streams to media requesters, such as other proxies or end users. Each AS has a multiplexing capacity that is limited by the processing power of the available proxy servers, or due to local bandwidth limitations. An AS may also serve as a "tunnel" passing a stream from one neighboring AS to another. (This will be explained further below in connection with Figure 8(b).) Although tunneling uses bandwidth resources at the peering points, it may not impact multiplexing capacity of the AS, which depends primarily on local resources.

[0027] In addition to the utilization costs, both monetary and in terms of network resources such as capacity, bandwidth, etc., purchasing, deploying and maintaining a proxy server in an AS is costly. Therefore, it is desirable for the overlay network to optimize the use of network resources and minimize the number of proxy servers used to handle a given workload. The invention takes these factors into consideration in establishing and maintaining the tree structures, and in servicing requests for access to streaming media. It does this in by the paths it establishes through a peer-to-peer negotiation process to service requests, as well as in allocating and using system resources. It is desirable to optimize the utilization of resources so that they are neither underutilized nor overutilized. Underutilization of resources is undesirable for cost reasons; similarly, overutilization of resources is undesirable for performance reasons. Overutilization can lead to network congestion, rejection of user requests for access, reduced stream quality, and vulnerability to sudden changes in system conditions, among other undesirable consequences. The invention addresses these issues by constructing and maintaining the multicast tree structures according to predetermined relationships between the load on system resources and the corresponding "tension" as a measure of cost for that load. The invention may employ different relationships for this purpose.

[0028] Figures 3(a) and (b) illustrate examples of different alternative tension versus load relationships which the invention may employ. These relationships are illustrated in the form of graphs that represent the tension as a measure of the different costs (both monetary and system resource costs) associated with utilization of proxy resources. Similar tension relationships may also be established for bandwidth resources. The relationships shown in Figures 3(a) and (b) show the cost associated with deviating from a preferred proxy resource utilization. They show that tension is minimized at a preferred or optimum level (t_{pref}) when a proxy resource is used at a preferred load (l_{pref}), and that tension increases sharply when its load is approaches maximum capacity (l_{limit}). Tension also increases to a local maximum (t_{l}) as the utilization (load) of the proxy drops below the preferred level, and tension is zero when the resource is not utilized at all.

[0029] As shown in Figure 3(a), tension decreases rather quickly from the local maximum (t_{l}) at a utilization level of 1 and flattens as utilization increases toward the preferred loading (l_{pref}) at which it is minimum (t_{pref}). As load continues to increase beyond the preferred level, tension again rapidly increases as the loading approaches the limit (l_{limit}) of the proxy server. The rapid increase in tension above the preferred loading in Figure 3(a) may represent in part, for example, the risks associated with overutilization of the resource, such as the previously mentioned congestion, rejection of user requests, reduced quality and vulnerability to system changes.

[0030] A different operating relationship that may be employed is shown in Figure 3(b). There, tension is flatter and decreases more slowly as loading increases from an initial loading, and then decreases rapidly as utilization approaches the preferred utilization level (l_{pref}). Thereafter, tension remains low above the preferred level as the load increases toward the limit of the resource (l_{limit}). Consequently, the relationship illustrated in Figure 3(b) indicates a preference for selecting proxies whose utilization is already close to the preferred level when choosing a proxy to service a new request, rather than selecting proxies that are more lightly loaded. Figure 3(b), moreover, reflects a policy that prefers selecting proxies that have loadings somewhat above the preferred level to satisfy new requests, and indicates a corresponding lack of concern for risks associated with overutilization. Both relationships shown in the figures show high tension is associated with using resources that are very lightly loaded; t_1 is the cost or tension of activating a new proxy in the network.

[0031] For a given set of proxies, a particular AS level network structure, a set of media sources, and particular loads accessing media, the invention creates multicast tree structures that optimize network resources and tension. More particularly, the invention creates structures such that resources are neither overutilized nor underutilized, and in which the overall operational cost is minimized. The end-to-end delays may also be small. It does this using an overlay network structure that employs proxy-to-proxy (peer-to-peer) negotiations for establishing and maintaining optimum resource utilization through redirection of requests, load redistributions, and load consolidations across ASs. As will be described in more detail below, when a proxy receives a join request for access to a media stream, the proxy may either admit (accept) the request, deny the request, or redirect the request to a more suitable proxy in its neighborhood. An overloaded resource may initiate a redistribution process in which it transfers part of its load to a different proxy. Similarly, an underutilized resource may request consolidation in which some load from an underloaded server may be transferred to another server. The invention provides protocols, as will be described, that enables the overlay network to scale and adapt itself as sources and requesters come and go and as multicast paths are created and destroyed. Unlike IP-multicasting-based approaches, the invention considers and integrates AS-level service agreements into its network structuring processes since it takes these into consideration as part of tension. Moreover, the invention maintains an awareness of the overall status of the network and seeks to optimize utilization of the network and proxy resources, along with an average delay, within an integrated framework. Unlike static overlay-network based approaches, the invention dynamically adapts to varying data, network and user load conditions in a distributed peer-to-peer manner to optimize the utilization and tension of network resources. The manner in which the invention accomplishes this will be described in more detail below.

[0032] Briefly summarized, the invention seeks to deliver media streams to end users using as few network resources as possible while preventing resource congestions and reductions in quality of service. It does this by deploying application-level hierarchical multicasting structures through its proxy servers, and so that peer-to-peer and overall network tension is minimized. The overall tension which is minimized comprises the sum of the tensions of peering points (btension_j) in the network, plus the sum of the proxy tensions (ptension_j) in the network. Significantly, the invention optimizes tension using a decentralized decision-making and adaptability approach. Decisions on establishing connections and loading are made through peer-to-peer negotiations between proxies. Moreover, as sources and requesters join and leave the overlay network, operating conditions are constantly changing. Since there is typically a significant cost associated with globally changing existing paths, the invention adapts to changes to the extent possible with local modifications.

[0033] The invention may employ several different complementary protocols for effecting an optimized overlay network multicast tree structure. These protocols comprise a media streaming protocol, an information exchange protocol, and a multicast management protocol. The media streaming protocol may be selected according to the type of media delivered through the media streaming overlay network of the invention. It specifies how media is transmitted from a source to an end user through a chain of proxies. Suitable protocols that may be used include, for example, Real Time Streaming Protocol (RTSP), Real Time Control Protocol (RTCP), and transport level protocols such as Real Time Transport Protocol (RTP). Other protocols may also be employed.

[0034] The information exchange protocol may comprise any suitable protocol by which a proxy communicates with other proxies and collects information about its environment and media sources. Table 1 (below) indicates some of the information which a proxy may collect from its neighbors and from the sources by communicating and exchanging status information with other proxies. Communication may be performed periodically, on a regular or irregular basis, or a proxy may initiate communications upon the occurrence of an event, such as a request for access. The regular information exchange helps proxies maintain information on network resource status, and identify when a connection or path between two proxies is interrupted. It may also declare a link failed when the quality of service decreases to an unacceptable threshold even though control messages may travel without problem across the link. Communication between proxies may be handled by a network monitoring process run on each proxy. In addition to periodic information exchange, each proxy may also attach a current list of its values to messages it exchanges with its neighbors, or explicitly request new information when it detects a change in status, as, for example, due to a failure or insertion of new sources.

Table 1: Information Collected and Stored by Proxy p_i

Information	**Meaning**
N_i	the list of proxies that are neighbors of p_i.
$bload_k$ $btension_k$	the current load and the tension relationship, respectively, of each path, e_k, from p_i to its neighbors
$pload_j$ $ptension_j$	the current load and the tension relationship, respectively, of each proxy p_j in its neighborhood
$minbcost_{i,j}$	the minimum cost of sending a stream from a source s_j to proxy p_i

[0035] In Table T1, a "neighbor" proxy refers to a proxy that is logically connected. Initially, it may comprise physical neighbors, i.e., proxies in an AS connected by a path to the AS of proxy p_i is a member. However, as load increases, the neighbor members may extend to include proxies that are not immediate neighbors.

[0036] Although most of the information in Table T1 requires knowledge about only the immediate neighborhood of a proxy, estimating $minbcost_{i,j}$, the minimum cost of send a stream from source s_j to proxy p_i based upon a current network tension and proxy utilization, requires more information on the multicast tree structure than that involving immediate neighbors. This information may be obtained by estimating source-to-proxy distances, either through the exchange of messages between proxies, or by assuming the shortest distance based on an assumption that the network and proxies are used at their preferred levels. This assumption is advantageous in that it eliminates the need for constant information exchange and instead requires communications only with respect to significant changes in network structure, such as the addition or removal of a proxy or a network edge.

[0037] The multicast management protocol is the protocol which embodies the peer-to-peer decision-making processes used for creating and managing the multicast tree structure. It also updates the structure as conditions such as load and request characteristics in the overlay network change. This protocol preferable operates in a request-driven manner, i.e., multicast tree structures are generated or changed as requests for access arrive or other conditions change. When a new source is inserted into the network, the only required initialization process is to make proxies aware of the new source. This may be achieved by a central lookup registry, which publishes a list of sources to the proxies, or, preferably, by allowing proxies to discover sources through the peer-to-peer information exchange protocol described above. Otherwise, the operation of each proxy is driven by join requests from other proxies, stop requests from other proxies, and messages that initiate changes in the multicast tree structures. The join and drop requests may result in load redirections, redistributions and consolidations in order to avoid overutilization or underutilization of a resource. Figures 4 - 6 illustrate, respectively, request redirection, load redistribution, and load consolidation processes.

[0038] Figure 4 illustrates two neighboring autonomous systems AS1 and AS2 having a common peering point or boundary 40. AS1 includes a proxy 42 receiving a media stream 44 from a corresponding source, and provides stream 44 to a child proxy 46. Stream 44 may consume approximately one-fourth of the capacity of proxy 42, for example, as indicated by the horizontal line and darkened lower one-quarter portion of the circle representing proxy 42 in Figure 4(a). As is also shown in Figure 4(a), AS2 may have a proxy 43 which receives a request 47 from a proxy 48 for access to media stream 44. Since proxy 43 is not receiving media stream

44, it seeks a more suitable proxy to serve the request, and enters negotiations via peering point 40 with proxy 42. Proxy 42 has excess capacity, and assuming the cost of supplying media stream 44 to proxy 48 would be less if supplied by proxy 42 rather than proxy 43, proxy 42 may admit the request from proxy 48 and begin supplying stream 44 to proxy 48. As indicated in Figure 4(b), this may increase the load on proxy 42 to approximately one-half of its capacity. Moreover, as shown in Figure 4(b), proxy 42 supplies stream 44 directly to proxy 48 via peering point 49. It is assumed the cost would be less than supplying the stream to first to proxy 43 via peering point 40 and then incurring an additional cost as proxy 43 supplies the stream to 48 via another peering point 50. This is determined by the peer-to-peer negotiations between proxies which seeks the minimum cost paths. The request redirection process illustrated in Figure 4 will be described more fully in connection with the description of Figure 7.

[0039] Figure 5 illustrates load redistribution to avoid overutilization of a proxy. As shown in Figure 5(a), a proxy 52 in AS1 may be receiving streaming media 53 and supplying the streaming media to proxies 54 and 55. As indicated in Figure 5(a), proxy 52 may be at substantially full capacity (as indicated by the completely darkened circle in the figure representing proxy 52). Therefore, it is operating at or close to its load limit, is being overutilized, and incurring an associated high tension, assuming the relationships indicated in Figures 3(a)-(b). Accordingly, proxy 52 will seek a suitable proxy in order to redistribute a portion of its load. As shown, it may enter negotiations with a neighboring proxy 56 to accept a portion of media stream 53. These negotiations will involve a comparison of costs or tension associated with transferring part of the load on proxy 52 to proxy 56. Since proxy 56 is unloaded and transferring the load incurred in supplying media stream 53 to proxy 55 may be assumed to take approximately one-half of the capacity of proxy 56 (see Figure 5(b)) this would result in proxies 52 and 56 operating at approximately their preferred load and incurring a corresponding preferred tension, as illustrated in Figure 3. Accordingly, proxy 56 establishes a connection to the source of media stream 53 and begins supplying the media stream to proxy 55 via a path 58.

[0040] Load redistribution to avoid overutilization, as shown in Figure 5, can result in more optimum utilization of network resources by reducing the tension on both proxies and paths.

[0041] Figure 6 illustrates a load consolidation process to avoid underutilization of a proxy. Referring to Figure 6(a), proxies 60 and 62 in neighboring autonomous systems AS1 and AS2 may be receiving media streams 63, 64, respectively, which may be the same or different media streams. Proxy 60 may be supplying media stream 63 to proxy 65 and proxy 62 may be supplying media stream 64 to proxy 66, as indicated in Figure 6(a). As also indicated in the figures, proxies 60 and 62 may be lightly loaded and, therefore, well below their preferred loads (l_{pref}). Accordingly, as shown in Figure 3, proxies 60 and 62 may both be incurring a relatively high tension, which they will discover through the information exchange protocol described previously. Thus, the two proxies may enter negotiations to determine the cost of consolidating their loads. Assuming that consolidating both loads in proxy 60, as shown in Figure 6(b), results in proxy 60 being loaded at approximately one-half of its capacity (assumed to be at its preferred load (l_{pref})), proxy 62 transfers its load to proxy 60, and proxy 60 begins to supply streaming media 64 to proxy 66 as indicated in Figure 6(b). Thus, by changing the multicast structure to consolidate loads, reduced tension and more optimum use of resources is afforded.

[0042] The restructuring processes of the multicast tree structure illustrated in Figures 4-6 provide an overview of the decentralized peer-to-peer negotiation processes of the invention which afford optimum utilization of network resources. Each of these will be described in more detail below.

[0043] Figure 7 isa flow chart of the process performed by a proxy in response to a join request by a requester, such as described above in connection with Figure 4. As shown, upon a proxy p_i receiving a join request 70 from either a requester, i.e., a user or another proxy, for access to a media stream multicast by a source s_j, the proxy has to determine whether it should admit the request or forward the request to another more suitable proxy. Proxy p_i first checks, at 72, to determine whether accepting the request 70 will cause a loop in the tree structure. Proxy p_i does this using the information exchange protocol previously described and the information collected and maintained by the proxy. If a loop is detected, the request may be redirected to another proxy p_j (at 74), as previously described, for example, in connection with Figure 4. If the request does not cause a loop, the proxy next checks, at 76, whether it has capacity to accept the request or whether accepting the request will possibly produce an overload. If an overload is possible, the proxy checks (at 77) whether redistribution of a part of its existing load to other proxy servers will enable it to accept the request. If so, the proxy redistributes a part of its load (at 78) and admits the request (at 80). If redistribution is not possible, the proxy then seeks a suitable proxy (at 82) in the neighborhood to serve the join request. If it locates one, the proxy may then redirect the request to that proxy (at 74). Otherwise, it denies the request at 84.

[0044] Even if proxy p_i has capacity to accept the request without the possibility of being overloaded, it nevertheless preferably checks (at 86) to determine whether there is a more suitable proxy p_j in the neighborhood to serve the join request. If so, it then redirects the request to proxy p_j (at 74). Proxy p_i checks (at 88) to determine whether it is already serving the stream. If so, then it may admit the request (at 80). If the proxy is not serving the stream, it next checks (at 90) for a more suitable neighbor, and redirects the request (at 74) to the neighbor. Otherwise, the proxy admits the request (at 80) and then sends a join request to a candidate parent (at 92) determined based upon network costs and tensions.

[0045] In step 76 of the process, the proxy first checked to determine whether accepting the request would result in a possible overload. Preferably, the invention does not define overload in absolute terms. Rather, the invention preferably defines an overload as a condition where redistributing a fraction (θ) of its load to some other neighboring proxy server, p_h, will be beneficial to the overall operation of the network, in terms of reduced tension, i.e., whether:

$$\text{t\_reduction\_at\_i} > (1 + \gamma_d) \text{ x } \text{t\_increase\_at\_h}$$

or, stated differently,

$$\text{ptension}_i (\text{pload}_i) - \text{ptension}_i ((1 - \theta) \text{ x } \text{pload}_i) >$$
$$(1 + \gamma_d) \text{ x } \text{ptension}_h (\theta) \text{ x } \text{pload}_i)$$

where, $0.0 \leq \gamma_d$ and γ_d is a threshold factor that may be selected to provide a level at which a load redistribution will be initiated. If this condition is satisfied, redirection or redistribution negotiations, as previously described, may occur between proxies to redistribute and handle loads.

[0046] Before admitting a join request for a stream from a source as described in connection with Figure 7, proxy p_i may check to determine whether there is any proxy in its neighborhood better suited to admit this request. Suitability is defined in terms of optimizing tension. The proxy p_i may do this check by first calculating a self-cost of admitting the request. If proxy p_i is already serving the stream, then the self-cost may be determined as the difference in tension between the new proxy tension if the request is admitted and the current proxy tension. This may be expressed as follows:

$$\text{self cost} = \text{ptension}_i\,(\text{new\_ pl}_i) - \text{ptension}_i\,(\text{old\_pl}_i)$$

[0047] If p_i is not serving the stream, the expected network cost (increase in the tension in the network resources) must also be taken into account. In this case, proxy p_i may estimate the cost of admitting this request as follows:

$$\text{self cost} = \text{ptension}_i(\text{new\_ pl}_i) - \text{ptension}_i(\text{old\_pl}_i) + \text{minbcost}_{i,j}$$

where $\text{minbcost}_{i,j}$ is the cheapest distance based on the current tension values in the network for a path between servers p_i and p_j. This value is only an estimate of the actual network tension and costs. The request for a particular stream may actually be routed through a more costly route depending on actual resource loads, or routing may cost much less if a multicast tree structure serving the stream is found nearby.

[0048] The costs of directing the request to the neighbors p_h of p_i may be estimated as:

$$\text{cost}_h = \text{ptension}_h(\text{new\_pl}_h) - \text{ptension}_h(\text{old\_pl}_h) + \text{minbcost}_{h,j} +$$
$$\text{btension}_{i,h}(\text{new\_bl}_{i,h}) - \text{btension}_{i,h}(\text{old\_bl}_{i,h})$$

The main difference in this cost estimation from the determination of the self cost is that in addition to estimating the cost for a neighboring proxy, p_h, p_i may also consider the cost associated with acting as a network tunnel for the request in the event that the requesting proxy does not have a direct connection to proxy p_h .

[0049] Figure 8 illustrates the redirection of a request and the tunneling process. As shown in Figure 8(a), a requesting proxy 100 that sends a request to a proxy p_i 102 in AS2 that is not serving a media stream 104 redirects that request to a neighboring proxy p_h 106 that is serving media stream 104, as previously described. Here, the requesting proxy 100 does not have a direct connection to proxy 106. In this event, proxy 102, which receives the request also considers the cost associated with acting as a network tunnel for the request. As a tunnel, proxy 102, which has a direct connection 108 to proxy 106, receives media stream 104 and tunnels or passes the media stream to requesting proxy 100. As shown in Figures 8(a) and (b), this increases the load on proxy 106, and this is taken into consideration in the above equation in determining the $cost_h$.

[0050] If the proxy originating the request and the neighbor proxy p_h both have a direct connection to a proxy serving the media stream, then that connection may be used after redirection. Also, in order to facilitate the redirection process, the cost estimation for neighbors in the network is preferably computed frequently by proxies (such as periodically when the information in Table T1 is updated) in anticipation of restructuring of the network. These may be kept in a local table at each proxy, such as Table 1. Once the cost estimates are computed, if the most suitable proxy is not the proxy p_i to which the join request was directed, that proxy p_i may send a list of candidate proxies and the associated costs which it determined to the requesting proxy along with a redirection message. The requesting proxy may then choose a proxy from the list that optimizes the network and forward a joint request to that proxy. Figure 9 illustrates the expansion of a neighborhood through redirection.

[0051] As shown in Figure 9(a), a proxy 110 may forward a request to a proxy 112 in its neighborhood 114, which includes proxies 116 and 118. Proxy 112 may be the proxy that was determined to be the most suitable candidate based upon cost estimates described above. As shown in Figure 9(b), the neighbor proxy 112 to which the original request was directed may choose to redirect the request to one of its neighbors 120, 122, either because of its own load or it knows that one of these neighbors is better suited for the request. Because of information exchanged between the proxies, the original proxy 110 now has a wider view of the neighborhood and can send the request to the most suitable proxy, e.g., 122, as shown in Figure 9(c). Thus, the peer-to-peer information exchange protocol enables proxies to become aware of the status of network resources and to adaptively access those resources in the most cost effective and optimum manner.

[0052] If a proxy p_i decides to admit a request, and if the request is for a stream that is not already served by this proxy, then p_i has to find a way to join the multicast tree structure that serves the stream. To do this, it evaluates its neighbors and selects the most suitable one to which to send a join request. As shown in Figure 10, a proxy p_i 130 may consider its neighbors 132, 134 in AS1 and AS2, respectively, when choosing the next proxy. For those neighbors such as p_l 132 that are already on the required multicast tree structure 140, p_i may estimate the cost of connection as:

$$\begin{aligned} cost_l = {} & ptension_l(new\_pl_l) - ptension_l\,(old\_pl_l) + \\ & btension_{il}\,(new\_bl_{i,l}) - btension_{il}\,(old\_bl_{i,l}) \end{aligned}$$

That is, proxy p_i, 130, accounts for the increased tension at the new parent (p_l) 132 as well as the increased tension on the network connection 142 between itself and parent 132. If a candidate parent proxy, 134, for example, is not serving the stream, proxy p_i 130 also accounts for the fact that the candidate parent 134 will need to establish a route 144 to the source, and determines the cost as:

$$\mathrm{cost}_l = \mathrm{ptension}_l(\mathrm{new\_pl}_l) - \mathrm{ptension}_l\,(\mathrm{old\_pl}_l) + \mathrm{btension}_{i,l}\,(\mathrm{new\_bl}_{i,l}) - \mathrm{btension}_{i,l}\,(\mathrm{old\_bl}_{i,l}) + \mathrm{minbcost}_{l,j}.$$

[0053] After the estimates are computed for all neighbors (132, 134), proxy p_i may choose the most suitable proxy and forward the request to that proxy. In order to prevent redirection loops, proxy p_i preferably maintains a list of the set of proxies to which it has already redirected requests. Once the neighbor proxy receives the join request, it can either deny, redirect, or admit the request, as described above.

[0054] A proxy p_i that has received a join request may either admit, deny, or redirect the request, as previously described, and may send back to the sending proxy a corresponding admit, deny or redirection message. Unless the request is admitted, the sending proxy has to find an alternative proxy to join the request. As described above, redirection messages preferably carry information about potential candidate proxies and their associated costs. The sending proxy may merge this information with the information it already has about its neighborhood and send the request to the most suitable candidate of which it is currently aware. Unless a limit on redirection requests is established, a request may be redirected multiple times. Therefore, preferably, a limit, $\mathrm{redir}_{\mathrm{limit}}$, is placed on the number of times each request can be redirected. Once the redirection limit is reached, if the proxy receiving the request fails to join to the stream, it may send a deny message downstream to the sending proxy waiting for the establishment of the connection. The downstream proxy may then initiate its own deny-message handling routines and take additional action, such as sending join requests to other proxies.

[0055] A join request may originate either from an edge proxy or from a proxy that is serving other proxies during establishment of the multicast tree structure. In the latter case, it is desirable to employ appropriate mechanisms to eliminate routing loops. A preferred way of accomplishing this in accordance with the invention is to annotate each join request with the name of the AS where the request originates. This is illustrated in Figure 11(a) where a request 150 from a proxy 152 in AS2 is labeled with its source (AS2) when it is sent to a receiving proxy 154. If the receiving proxy already has a path to the source 156 of the stream, it only has to verify that the list it maintains of its parents does not include AS2 where the join request originated. If there is not already an established path to the source, then the receiving proxy may take steps to ensure that the path 158 (see Figure 11(b)) that it will create to the source does not contain AS2. To achieve this, each join request is also preferably annotated with the list of downstream proxies waiting for the establishment of a path to the source. Consequently, a proxy receiving a request will be able to easily check whether admitting the request will cause a loop.

[0056] When a proxy receives a stop request from a user or from another proxy for access to the media stream multicast at a source, it may simply drop the requesting proxy from the multicast tree structure and makes the corresponding resources available for future requests. If there are no other child proxies consuming a stream served by the proxy, it may also sends a stop request to the corresponding parent for that stream so that it can release all corresponding resources for other use.

[0057] One of the scalability mechanisms used by the proxies of the invention is redistribution. When proxy p_i detects any change in its neighborhood (for example, as a result of the update messages it exchanges with its neighbors), such as a proxy becoming available or a configuration change, the proxy may check whether if it is overloaded or underloaded relative to its neighbors. Overload may be defined, as previously described, as a condition where the reduction in tension resulting from redistribution of a stream from proxy p_i is greater than a predetermined increase in tension by redistribution of the stream to a neighbor proxy p_h as determined by a threshold factor. This may be expressed as:

$$\text{t\_reduction\_at\_i} > (1 + \gamma_d)\text{t\_increase\_at\_h}$$

Here, γ_d is a threshold factor selected to have a value to prevent small gains in tension reduction from causing potentially costly redistributions. If proxy p_h is under-utilized, redistribution can also decrease the tension at p_h benefiting both of the involved proxies. If proxy p_i notices that this trigger condition is true for at least one of the streams, then it may initiate a redistribution process seeking to rebalance the overall tension in the system.

[0058] During the redistribution process, a proxy p_i seeks proxies in its neighborhood that can take a fraction of its load, e.g., 50% in a preferred embodiment. It may first choose the stream, s_j, whose redistribution will bring the highest benefit to the system. Then, for this stream, it may choose a proxy, p_l, to admit the fraction of the load it seeks to redistribute. The process that proxy p_i uses to choose proxy p_l may be similar to the process described previously for choosing the most suitable proxy to which to redirect a request, except that during the redistribution more than one connection may be redirected simultaneously. Hence, the loads that are used for calculating tension changes are based on the fraction of the load being shipped.

[0059] Once proxy p_i chooses the stream s_j and the proxy p_l to which to redistribute the load, proxy p_i sends a message to proxy p_l requesting it take the required load. In response, proxy p_l can either admit this load request, deny it, or redirect it. Before admitting the request proxy p_l may ensure that there is a loop free path to the source as shown in Figure 12. As in the case of an individual join request, if proxy p_l chooses to redirect the shipment request, redirection messages are preferably accompanied by the neighborhood list to help proxy p_i choose the most suitable proxy in its own and p_l's neighborhood.

[0060] Once a proxy p_l accepts the shipment, p_i starts shipping the load. During this process it preferably locks all resources (hence can not admit new requests). Proxy p_i may choose a fraction of its children which are consuming the streams s_j and redirect each one of them individually to proxy p_l. Proxy p_l may handle these join requests as a group and admit all of them.

[0061] Once the processing for stream s_j is complete, proxy p_i may choose another stream and continue this process until it redirects the required fraction of the load. Once a redirection to a proxy fails (for instance due to link failures), a timeout mechanism may be used to prevent more time being wasted trying to connect to it in the future.

[0062] Figure 12 illustrates the redistribution process and the avoidance of loops. As shown in Figure 12(a), a requesting proxy server 170 may send a list of the immediate children proxies 171-174 in autonomous systems AS2, AS7, AS3, and AS8, respectively, to the chosen proxy 176 for redistribution. If proxy 176 is able to establish a path 178 to a source 180 that does not go through any other children, as shown in Figure 12(b), then the proxy can redirect any set of the children, such as 173-174 to source 180 through path 178. However, if, as shown in Figure 12(c), a path 182 is created by the proxy 176 that passes through a child, such as 172, then a set of proxies that does not include proxy 172 may be redistributed.

[0063] A proxy p_i may decide to request consolidation in its neighborhood if it determines that a underloaded condition exists as a result of update messages that it exchanges with its neighbors. In this event, the proxy attempts to consolidate its service with that of its neighbors. Consolidation may be triggered when there is at least one proxy server p_h in the neighborhood such that if it ships one of its streams to proxy p_h the reduction in tension is greater than a predetermined factor times the increase in tension which would be produced at proxy p_h. This may be expressed as follows:

$$t\_reduction\_at\_i > (1 + \gamma_c)t\_increase\_at\_h$$

where γ_c is a threshold factor selected to have a value that avoids small gains in tension reduction that may produce potentially costly consolidations. Consequently, the consolidation process may be very similar to the redistribution process described above except that the amount of the load negotiated between proxies and redistributed from one proxy to another is the entire load for each stream instead a portion of the load. Upon consolidation of all streams for proxy p_i, the proxy becomes unutilized and may be taken out of the network and reserved for future use.

[0064] In spite of apparent similarities between consolidations and redistributions, it is easier to predict when a network needs consolidations than when it needs redistributions. This is because consolidations address a current loading where network resources are underutilized, whereas redistributions seek to ensure available resources for an anticipated future situation.

[0065] In the event of a failed connection between a proxy and its parent for a stream it is serving, the proxy may engage in a process to repair the failed connection that is similar to redirection. The proxy may redirect itself to another proxy in the neighborhood based on the network and proxy tensions, as previously described, to substitute for the failed connection. A timeout mechanism may be employed in order to avoid having the failed parent being used for a predetermined period of time, and the proxy may lock all resources so that they may not admit new requests for the failed stream.

[0066] From the foregoing, it can be seen that the invention employs different types of tension relationships in creating multicast tree structures. These relationships relate to proxy tension and bandwidth tension of the connections between proxies. Although the relationships relate to different resources, they are somewhat related. First, irrespective of internal resources, a proxy server cannot multiplex more streams than permitted by its outgoing peering points. Although an autonomous system can tunnel streams between other autonomous systems without multiplexing, this still requires the use of peering resources and the capacity of these resources by the multiplexing power of the proxy.

[0067] Moreover, the cost of the network may have a significant impact on redirection and redistribution. If redirection through network tunneling is less expensive, it may be preferable to activate proxies close to the edge of the network and use redirections instead of redistributions. This may result in shallow multicast tree structures with long proxy-to-proxy network connections where most edge proxies are connected to proxies that are closer to the source. Similarly, the cost of network links in relation to the cost of activating proxies will have an impact on the use of redistributions, and the cost of creating a new proxy, the cost of redirection, and the depth of the multicast restructure are likewise related and influence the creation of multicast tree structures.

[0068] While the foregoing has been with reference to certain preferred embodiments of the invention, it will be appreciated that these embodiments may be changed without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.